

Contrôle continu

Durée 1h30. Documents autorisés : une feuille A4 R/V notes de CM et de TP.

Nom et prénom :

Groupe :

Exercice 1 Que font les commandes suivantes ?

1. `find /bin -perm -110 -depth 1 -name 'p?'`
cherche dans le répertoire `/bin` et à une profondeur de 1 les fichiers dont le nom commence par 'p' et est suivi d'un seul caractère et ayant au minimum les droits d'exécution pour le propriétaire et son groupe.
2. `ls / | head -5`
liste les 5 premiers fichiers du répertoire racine '/' (alphabétiquement les premiers).
3. `ls exo[0-9]-[1-5]? .sh`
liste les fichiers script shell dont le nom commence par `exo`, suivi par un chiffre et un tiret -, puis un chiffre de 1 à 5, puis un caractère quelconque. Le nom du script doit avoir l'extension `.sh`
4. `sort -r < texte1.txt > texte2.txt`
trie le contenu du fichier `texte1.txt` en lignes entières dans l'ordre lexicographique décroissant et met le résultat dans le fichier `texte2.txt`. Si le fichier `texte2.txt` existait son contenu sera remplacé.

NB : l'option `-perm -mode` de la commande `find` sélectionne les fichiers ayant au minimum les droits spécifiés par `mode`.

Exercice 2 Donner le résultat de la commande :

```
for x in {5..10}; do let s+=x; done; echo $s;
```

affiche la valeur 45 ($\sum_{i=5}^{10} i$)

Exercice 3 La commande `sleep` permet de marquer un temps d'arrêt dans un script bash (`sleep - suspend execution for an interval of time`). Considérons le script `narco.sh` suivant :

```
#!/bin/bash
for x in $*; do
  ( sleep $x; echo -n $x) &
done
```

Quel est le résultat de `narco.sh 3 1 4 1 5` ?

11345

Exercice 4 La commande `date` affiche la date courante au format suivant :

Lun 18 oct 2021 21:59:17 CEST

Écrire une commande bash en une ligne qui affiche seulement l'heure et les minutes comme suit :

21:59

Réponse : `date | tr -s ' ' | cut -d' ' -f5 | cut -d':' -f1,2`

Exercice 5 Rappelons que la commande `ls -l` produit un affichage comme suit pour chaque fichier ou répertoire :

```
-rw-rw-r-- 1 toto  staff  239 16 oct  2020 exo1-tp3.sh
-rwx--x--x 1 toto  staff  222 27 sep  2020 exo1.sh
-rw-r--r-- 1 toto  staff  327 16 oct  2020 exo2.sh
-rwx--x--x 1 toto  staff  197 15 déc  2020 exo3-1.sh
```

Écrire une commande bash en une ligne qui compte le nombre de fichiers script shell accessibles en lecture pour les utilisateurs du même groupe que le propriétaire.

Réponse : `ls -l | cut -c5 | sort | uniq -c | tail -1`

Exercice 6 Soit un fichier dont le nom est dans une variable `fnom`. Écrire un script shell qui crée un fichier dont le nom est le contenu de la variable `fnom` auquel on a ajouté le suffixe `2` (par exemple, si le fichier s'appelle `foo`, le fichier qu'on doit créer s'appellera `foo2`) et tel que le contenu de ce fichier créé est le même que celui du fichier d'origine sans aucune ligne vide.

```
#!/bin/bash

fnom="foo"
taille=$( wc -l < "$fnom")

for i in $( seq 1 $taille); do
    ligne=$( cat "$fnom" | head -n $i | tail -n 1)
    lg=${#ligne}
    if [ $lg -ne 0 ]; then
        echo $ligne >> $fnom"2"
    fi
done
```

Exercice 7 Soit la commande qui déplace tous les fichiers dont le nom finit par `".mov"` dans un répertoire fils déjà existant appelé `films`. Écrire un script shell qui réalise cette commande en adoptant les deux solutions suivantes :

1. utiliser tout simplement l'instruction `for fichier in *.mov`

```
#!/bin/bash

for nom in *.mov; do
    mv $nom films
done
```

2. utiliser la commande `find`, sans descente récursive (avec une profondeur maximale de 1).

NB : avec l'option `-name`, la commande `find` permet de chercher des noms de fichiers spécifiques.

À partir du répertoire courant, la commande `find` suffixe par `'./'` les noms de fichiers. On rappelle que la longueur d'une chaîne `s` est obtenue par `${#s}` et que `${s : position}` désigne la chaîne extraite de `s` à partir de l'index `position`.

```
#!/bin/bash

find . -name "*.mov" -maxdepth 1 -print > /tmp/temp

taille=$( wc -l < /tmp/temp)

for i in $( seq 1$taille); do
    nom=$( cat /tmp/temp | head -n $i | tail -n 1)
    lg=${#nom}
    nom1=${nom:2}
    mv $nom1 films
done
```

Exercice 8 Un enseignant de TD dispose d'un fichier `eleves.txt` contenant les prénoms et noms de l'ensemble des étudiants du groupe de TD. Le fichier est de la forme suivante :

```
nom1:prenom1
nom2:prenom2
nom3:prenom3
```

1. Écrire un script shell qui crée un fichier `emails.txt` dans lequel chaque ligne contient l'adresse mail d'un étudiant du fichier `eleves.txt`, sous la forme `prenom.nom@etu.univ-orleans.fr`.

```
fnom="eleves.txt"

taille=$( wc -l < $fnom)
for i in $( seq 1 $taille); do
    ligne=$( cat "$fnom" | head -n $i | tail -n 1)
    prenom=$( echo $ligne| cut -d':' -f 2)
    nom=$( echo $ligne| cut -d':' -f 1)
    adr=$prenom'.'$nom'@etu.univ-orleans.fr'
    echo $adr >> emails.txt
done
```

2. En utilisant le fichier `emails.txt`, écrire *une commande bash* en une ligne de commande pour afficher la liste des étudiants sous le format

```
prenom1 nom1
prenom2 nom2
prenom3 nom3
```

Réponse : `cat emails.txt | cut -d'@' -f1 | tr '.' ' '`

Exercice 9 On possède un fichier `IP.csv` qui contient les informations sur l'inscription pédagogique, où chaque ligne est du format suivant :

```
nom:prenom:inscIF01:inscIF02:inscIF03:inscIF04:inscIF05
```

où pour les cinq modules, `inscIF0i` est *oui* ou *non* pour l'inscription dans le module.
Écrire un script shell qui compte le nombre d'inscriptions pour le module IF03.

```
fnom="IP.csv"
n=0
taille=$( wc -l < $fnom)
for i in $( seq 1 $taille); do
    ligne=$( cat "$fnom" | head -n $i | tail -n 1)
    insc=$( echo $ligne| cut -d':' -f 5)
    if [ "$insc" != "non" ] ; then
        n=$((n+1))
    fi
done
echo $n
```